10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Decision Trees

Matt Gormley
Lecture 2
August 29, 2018

SYLLABUS HIGHLIGHTS

Syllabus Highlights

The syllabus is located on the course webpage:

http://www.cs.cmu.edu/~mgormley/courses/10601-s18

The **course policies** are **required** reading.

Syllabus Highlights

- **Grading**: 45% homework, 25% midterm exam, 30% final exam
- **Midterm Exam**: evening exam, October 25, 2018
- **Final Exam**: final exam week, date TBD
- **Homework**: ~5 written and ~5 programming
  - 6 grace days for programming assignments only
  - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
  - No submissions accepted after 4 days w/o extension
  - Extension requests: see syllabus
- **Recitations**: Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings**: required, online PDFs, recommended for after lecture
- **Technologies**: Piazza (discussion), Autolab (programming), Canvas (quiz-style), Gradescope (open-ended)
- **Academic Integrity**:
  - Collaboration encouraged, but must be documented
  - Solutions must always be written independently
  - No re-use of found code / past assignments
  - Severe penalties (i.e.. failure)
- **Office Hours**: posted on Google Calendar on "People" page

Reminders

- **Homework 1: Background**
  - **Out: Wed, Aug 29**
  - **Due: Wed, Sep 05 at 11:59pm**
  - Two parts:
    1. written part to Gradescope,
    2. programming part to Autolab
  - unique policy for this assignment:
    1. **two submissions** for written (see writeup for details)
    2. **unlimited submissions** for programming (i.e. keep submitting until you get 100%),
  - unique policy for this assignment: we will grant (essentially) any and all extension requests

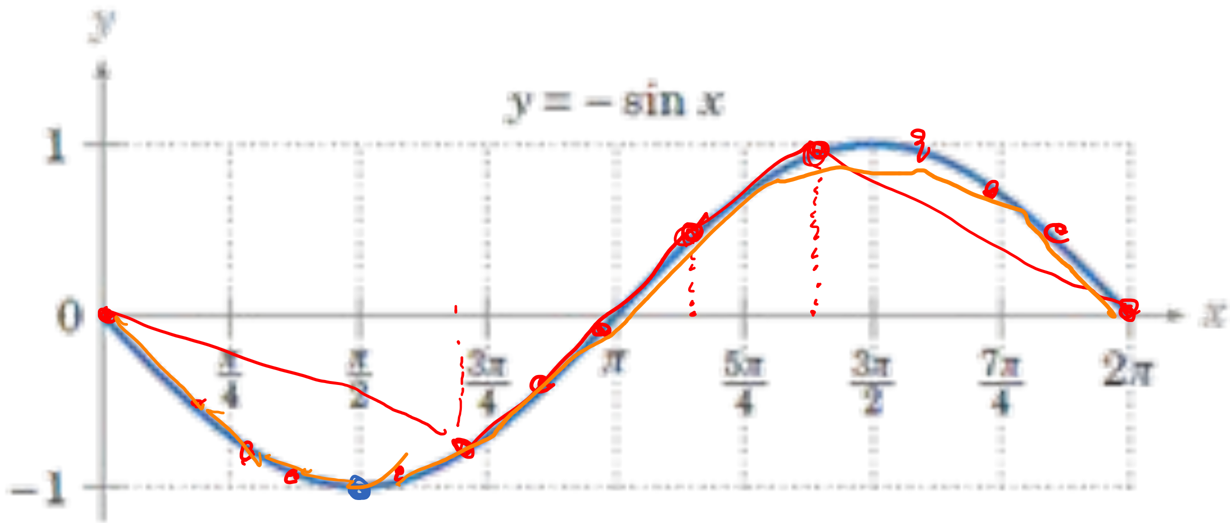
Big Ideas

1. How to formalize a learning problem
2. How to learn an expert system (i.e. Decision Tree)
3. Importance of inductive bias for generalization
4. Overfitting

FUNCTION APPROXIMATION

Function Approximation

Quiz: Implement a simple function which returns sin(x).



A few constraints are imposed:

1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of sin(x) a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in [0, 2*pi]

Medical Diagnosis

- Setting:
  - Doctor must decide whether or not to prescribe a treatment
  - Looks at attributes of a patient to make a medical diagnosis
  - Prescribes treatment if diagnosis is positive
- Key problem area for Machine Learning
- Potential to reshape health care

ML as Function Approximation

Chalkboard

- ML as Function Approximation
  - Problem setting
  - Input space
  - Output space
  - Unknown target function
  - Hypothesis space
  - Training examples

DECISION TREES

Decision Trees

Chalkboard

- Example: Medical Diagnosis
- Does memorization = learning?
- Decision Tree as a hypothesis
- Function approximation for DTs
- Decision Tree Learning

Tree to Predict C-Section Risk

Learned from medical records of 1000 women (Sims et al., 2000)

Negative examples are C-sections

```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+ .
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Figure from Tom Mitchell

Decision Trees

Chalkboard

- Decision Tree Learning
- Information Theory primer
  - Entropy
  - (Specific) Conditional Entropy
  - Conditional Entropy
  - Information Gain / Mutual Information
- Information Gain as DT splitting criterion

Tennis Example

Dataset:

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure from Tom Mitchell

Tennis Example

Which attribute yields the best classifier?

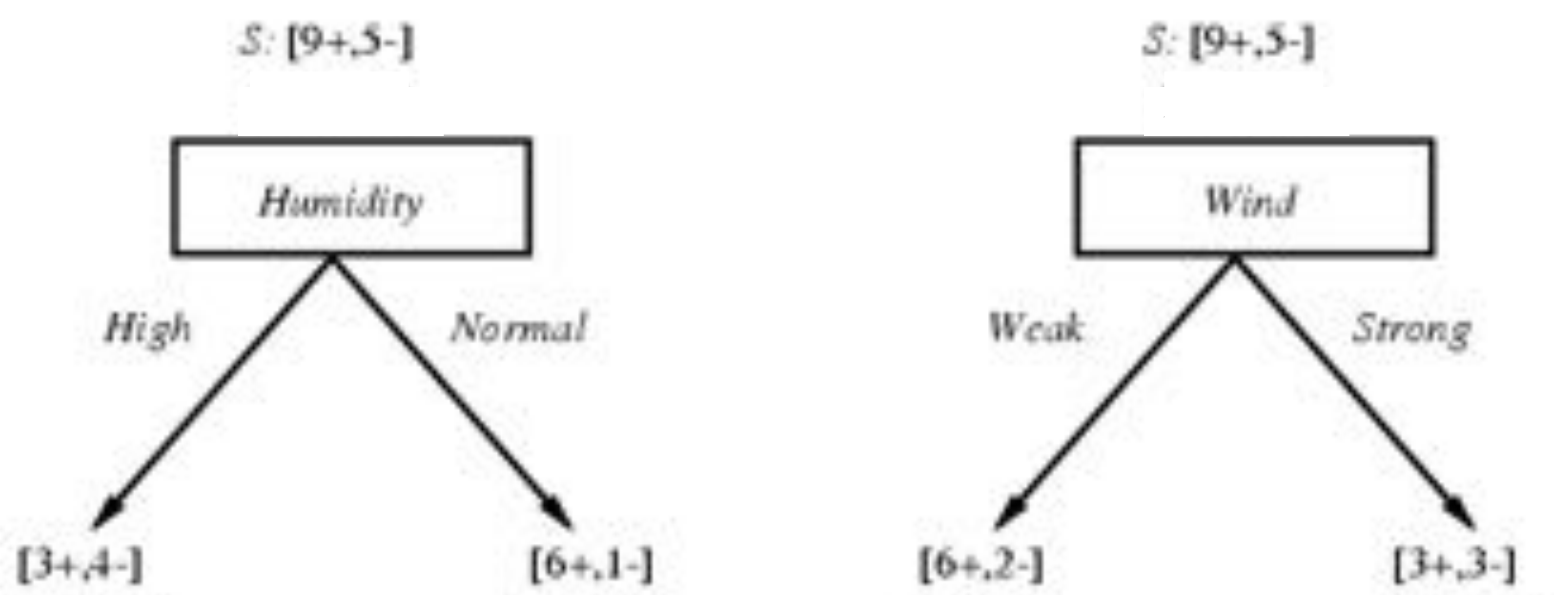


Figure from Tom Mitchell

Tennis Example

Which attribute yields the best classifier?

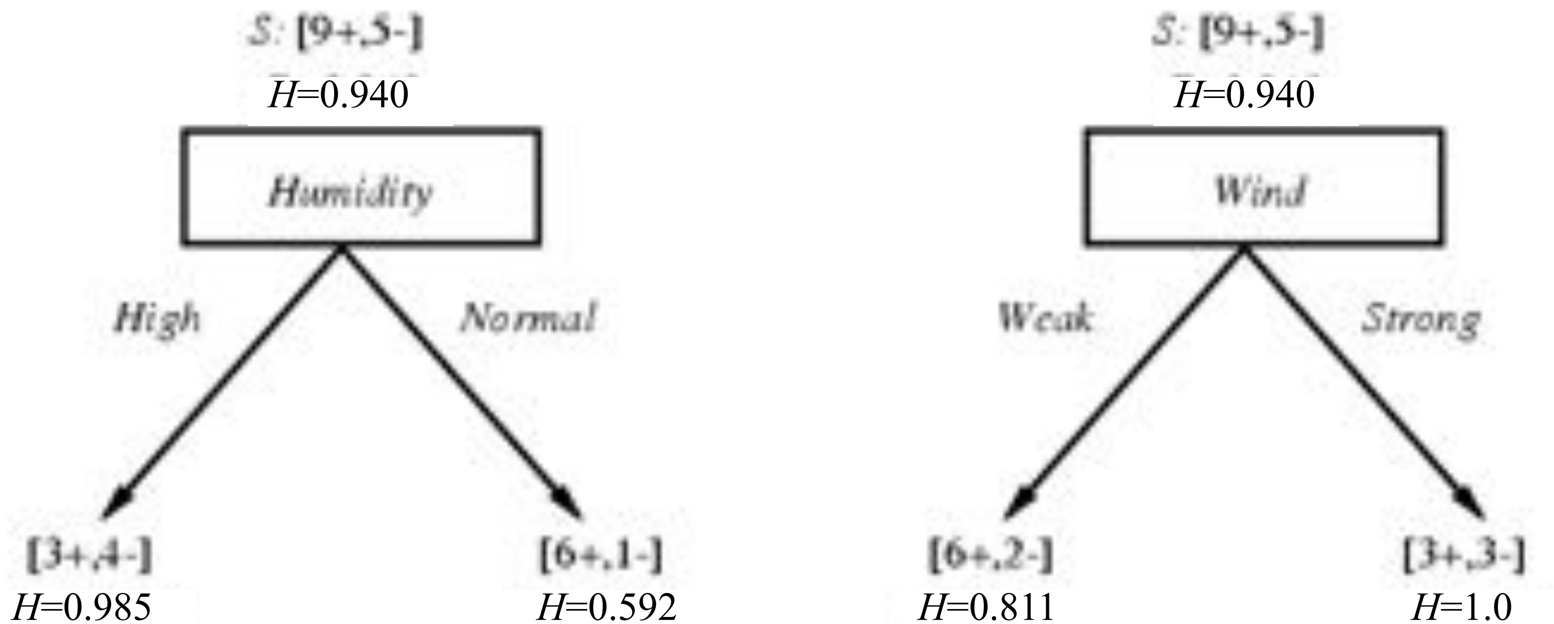


Figure from Tom Mitchell

Tennis Example

Which attribute yields the best classifier?

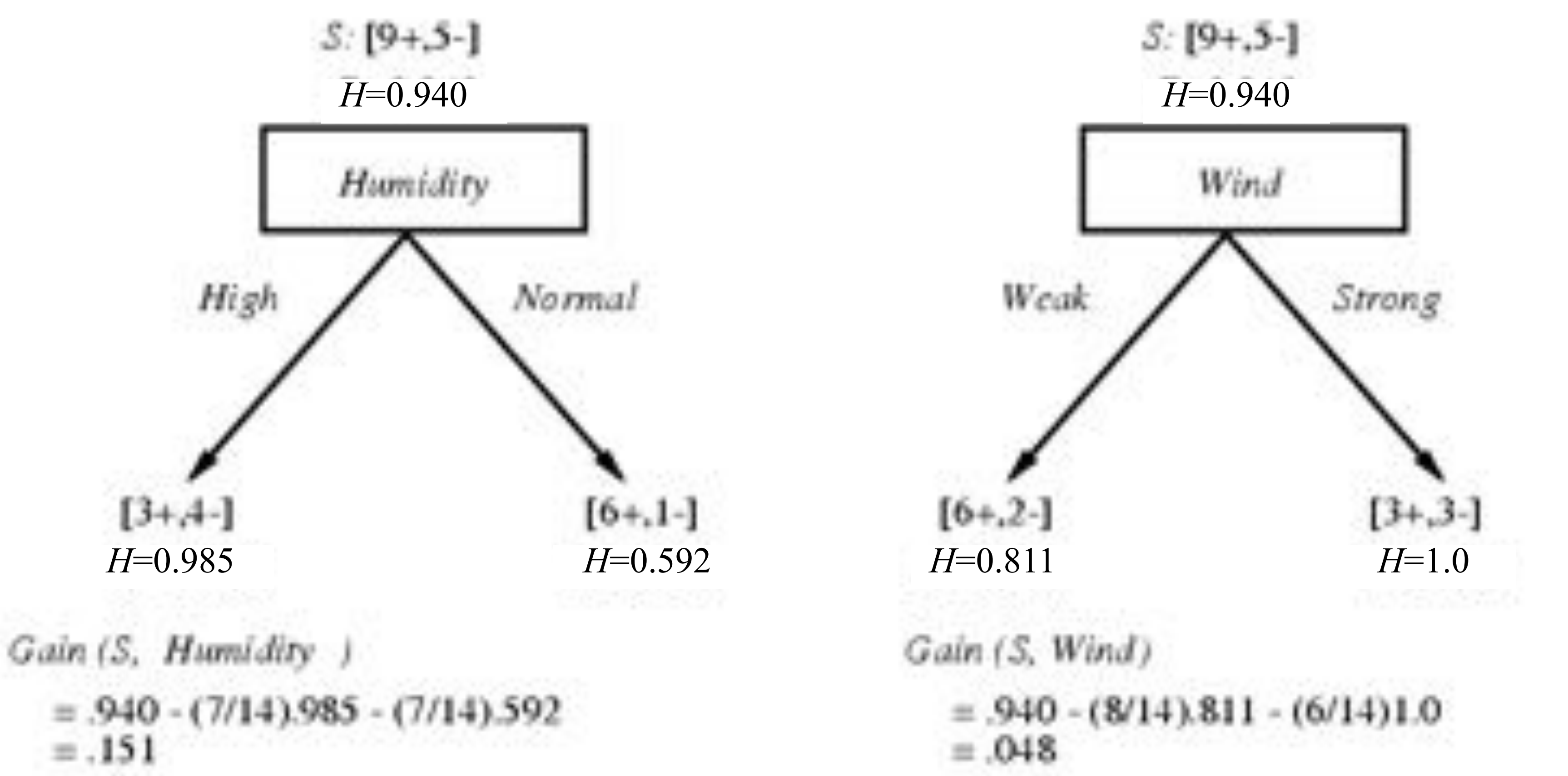
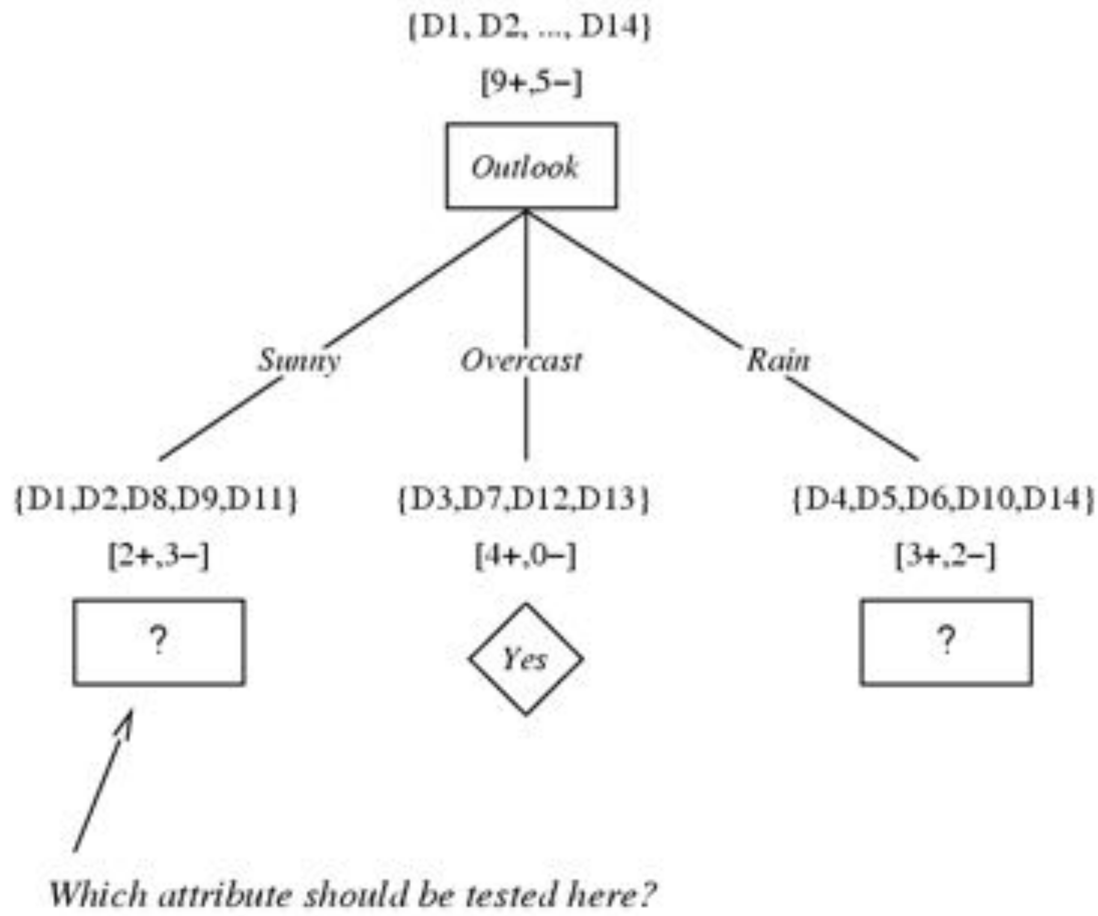


Figure from Tom Mitchell

Tennis Example



S_{sunny} = {D1,D2,D8,D9,D11}

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

Figure from Tom Mitchell

Decision Tree Learning Example

Dataset:

Output Y, Attributes A and B

Y	A	B
0	1	0
0	1	0
1	1	0
1	1	0
1	1	1
1	1	1
1	1	1
1	1	1

In-Class Exercise

1. Which attribute would **misclassification rate** select for the next split?
2. Which attribute would **information gain** select for the next split?
3. *Justify your answers.*

Decision Tree Learning Example

Dataset:

Output Y, Attributes A and B

Y	A	B
0	1	0
0	1	0
1	1	0
1	1	0
1	1	1
1	1	1
1	1	1
1	1	1

Decision Trees

Chalkboard

- ID3 as Search
- Inductive Bias of Decision Trees
- Occam's Razor

Overfitting and Underfitting

Underfitting

- The model…
  - is too simple
  - is unable captures the trends in the data
  - exhibits too much bias
- *Example*: majority-vote classifier (i.e. depth-zero decision tree)
- *Example*: a toddler (that has **not** attended medical school) attempting to carry out medical diagnosis

Overfitting

- The model…
  - is too complex
  - is fitting the noise in the data
  - or fitting random statistical fluctuations inherent in the "sample" of training data
  - does not have enough bias
- *Example*: our "memorizer" algorithm responding to an "orange shirt" attribute
- *Example*: medical student who simply memorizes patient case studies, but does not understand how to apply knowledge to new patients

Overfitting

Consider a hypothesis h and its

- Error rate over training data: $error_{train}(h)$
- True error rate over all data: $error_{true}(h)$

We say h <u>overfits</u> the training data if

$$error_{true}(h) > error_{train}(h)$$

Amount of overfitting =

$$error_{true}(h) - error_{train}(h)$$

Overfitting in Decision Tree Learning

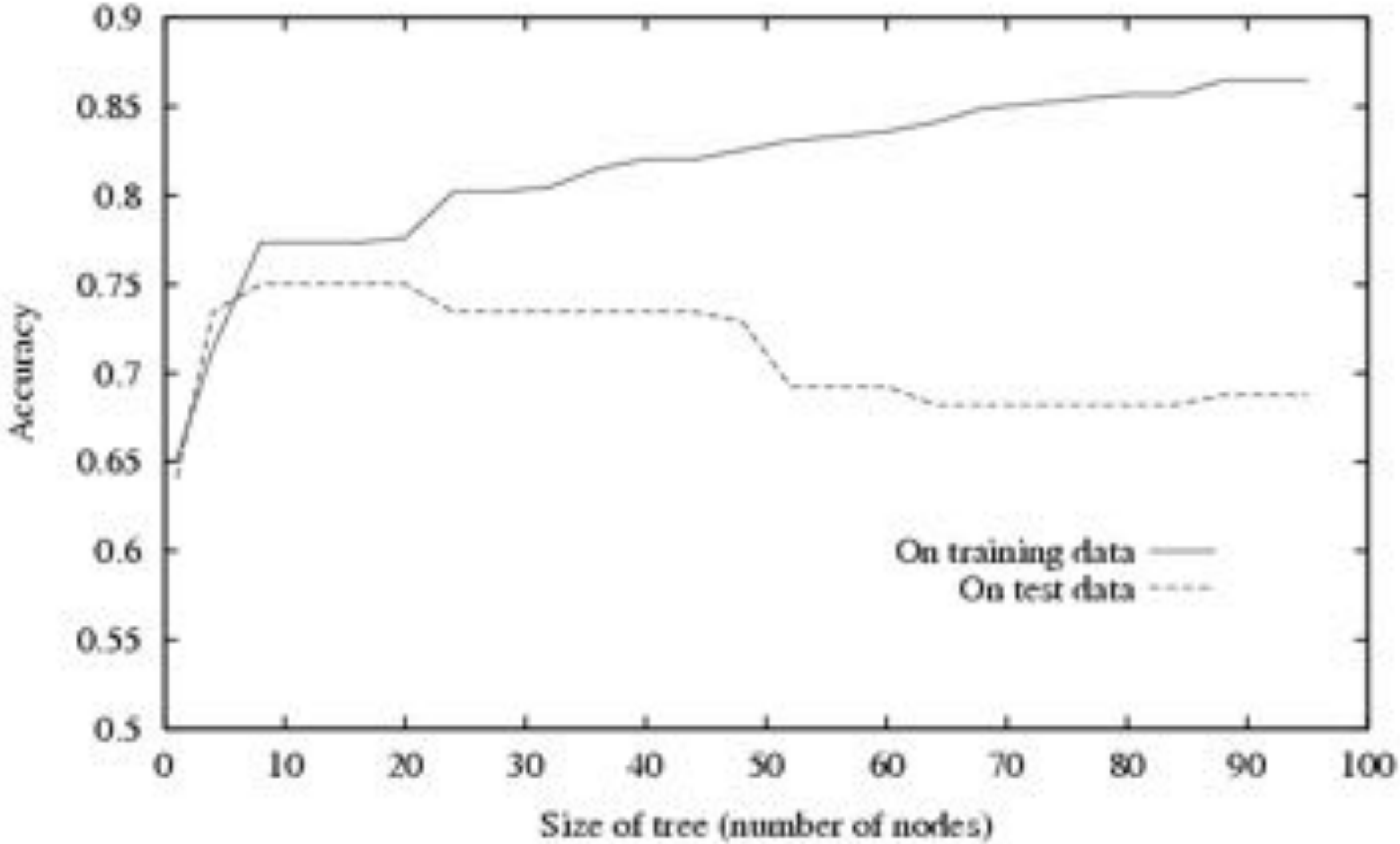


Figure from Tom Mitchell

How to Avoid Overfitting?

For Decision Trees…

1. Do not grow tree beyond some **maximum depth**
2. Do not split if splitting criterion (e.g. Info. Gain) is **below some threshold**
3. Stop growing when the split is **not statistically significant**
4. Grow the entire tree, then **prune**

Reduced-Error Pruning

Split data into *training* and *validation* set

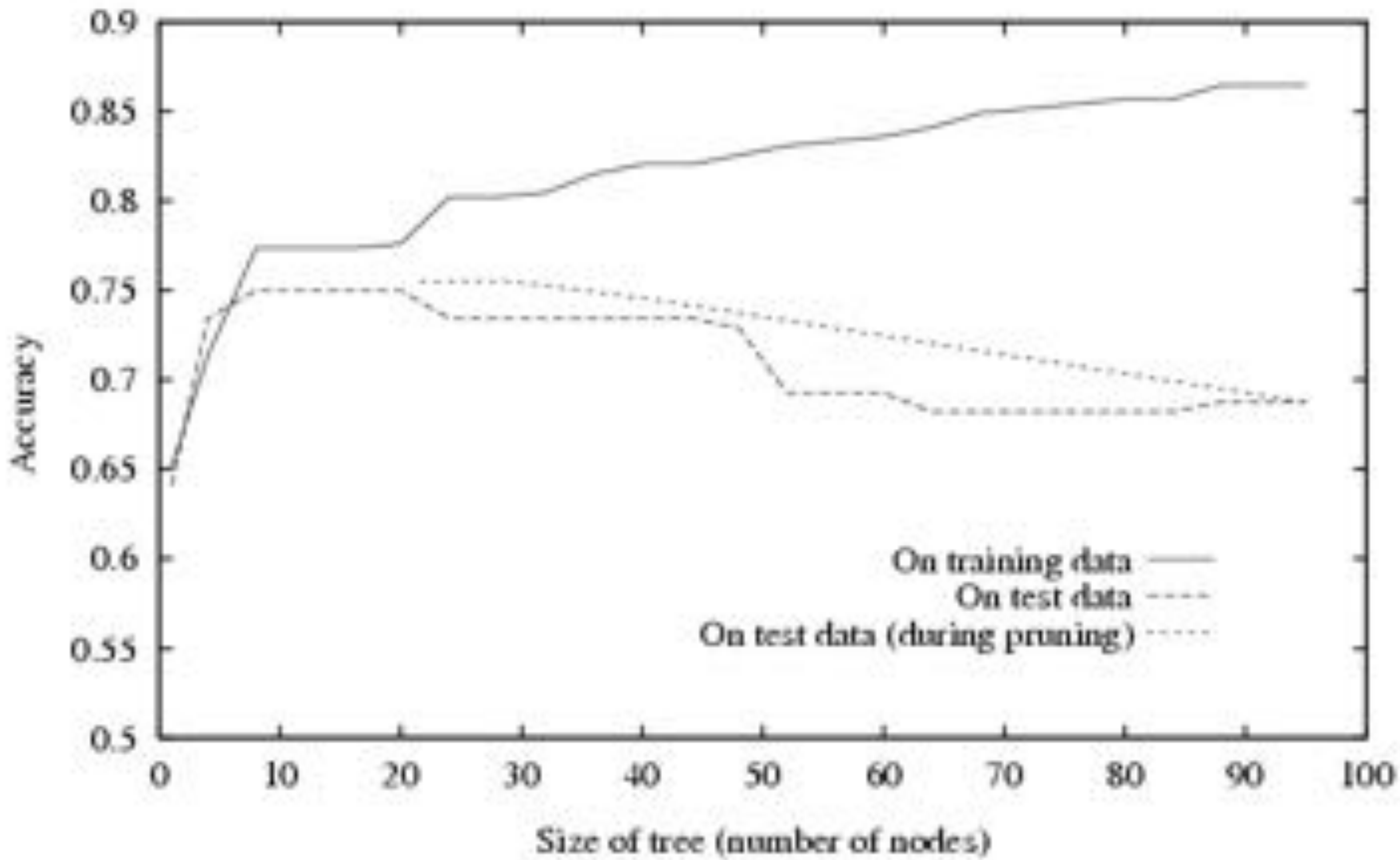
Create tree that classifies *training* set correctly

Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?

Effect of Reduced-Error Pruning



Slide from Tom Mitchell

Questions

- Will ID3 always include all the attributes in the tree?
- What if some attributes are real-valued? Can learning still be done efficiently?
- What if some attributes are missing?

Decision Trees (DTs) in the Wild

- DTs are one of the most popular classification methods for practical applications
  - Reason #1: The learned representation is **easy to explain** a non-ML person
  - Reason #2: They are **efficient** in both computation and memory
- DTs can be applied to a wide variety of problems including **classification, regression, density estimation,** etc.
- **Applications of DTs** include…
  - medicine, molecular biology, text classification, manufacturing, astronomy, agriculture, and many others
- **Decision Forests** learn many DTs from random subsets of features; the result is a very powerful example of an **ensemble method** (discussed later in the course)

DT Learning Objectives

You should be able to…

1. Implement Decision Tree training and prediction
2. Use effective splitting criteria for Decision Trees and be able to define entropy, conditional entropy, and mutual information / information gain
3. Explain the difference between memorization and generalization [CIML]
4. Describe the inductive bias of a decision tree
5. Formalize a learning problem by identifying the input space, output space, hypothesis space, and target function
6. Explain the difference between true error and training error
7. Judge whether a decision tree is "underfitting" or "overfitting"
8. Implement a pruning or early stopping method to combat overfitting in Decision Tree learning